Some Results on Learning

B.K. Natarajan

CMU-RI-TR-89-6

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

February 1989

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CMU-RI-TR-89-6	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION The Robotics Institute Carnegie Mellon University 6b OFFICE SYMBOL *(If applicable)*	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS *(City, State, and ZIP Code)* Pittsburgh, PA 15213	7b. ADDRESS *(City, State, and ZIP Code)*
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS *(City, State, and ZIP Code)*	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.
11. TITLE *(Include Security Classification)* Some Results on Learning	
12. PERSONAL AUTHOR(S) B.K. Natarajan	
13a. TYPE OF REPORT Technical 13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT *(Year, Month, Day)* February 1989 15. PAGE COUNT 25
16. SUPPLEMENTARY NOTATION	

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
algorithms, sets, functions

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This paper presents some formal results on learning. In particular, it concerns algorithms that learn sets and functions from examples. We seek conditions necessary and sufficient for learning over a range of probabilistic models for such algorithms.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE *(Include Area Code)* / 22c. OFFICE SYMBOL

Table of Contents

Abstract

This paper presents some formal results on learning. In particular, it concerns algorithms that learn sets and functions from examples. We seek conditions necessary and sufficient for learning over a range of probabilistic models for such algorithms.

1. Introduction

This paper concerns algorithms that learn sets and functions from examples for them. The results presented in this paper appeared in preliminary form in [Natarajan, 1986; 1988]. The motivation behind the study is a need to better understand the class of problems known as "concept learning problems" in the Artificial Intelligence literature.

What follows is a brief definition of concept (or set) learning. Let Σ be the (0,1) alphabet, Σ^* the set of all strings on Σ, and for any positive integer n, Σ^n the set of strings on Σ of length n. Let f denote a subset of Σ^* and F a set of such subsets. An example for f is a pair (x,y), $x \in \Sigma^*$, $y \in \Sigma$, such that $x \in f$ iff $y=1$. Informally, a learning algorithm for F is an algorithm that does the following: given a sufficiently large number of randomly chosen examples for any set $f \in F$, the algorithm identifies a set $g \in F$, such that g is a good approximation of f. (These notions will be formalized later.) The primary aim of this paper is to study the relationship between the properties of F and the number of examples necessary and sufficient for any learning algorithm for it.

To place this paper in perspective: There are numerous papers on the concept learning problem in the artificial intelligence literature. See [Michalski et al., 1983] for an excellent review. Much of this work is not formal in approach. On the other hand, many formal studies of related problems were reported in the inductive inference literature. See [Angluin & Smith, 1983] for an excellent review. As it happened, the wide gap between the basic assumptions of inductive inference on the one hand, and the needs of the empiricists on the other, did not permit the formal work significant practical import. More recently, [Valiant, 1984] introduced a new formal framework for the problem, with a view towards probabilistic analysis. The framework appears to be of both theoretical and practical interest, and the results of this paper are based on it and its variants. Related results appear in [Angluin, 1987; Rivest & Schapire, 1987; Berman & Roos, 1987; Laird, 1986; Kearns et al., 1986] amongst others. [Blumer et al., 1986] present an independent development of some of the results presented in this paper, their proofs hinging on some classical results in probability theory, while ours are mostly combinatorial in flavour.

We begin by describing a formal model of learning, our variant of the model first presented by [Valiant, 1984]. Specifically, we define the notion of polynomial learnability of sets in Section 2. We then discuss the notion of asymptotic dimension of a family of concepts, and use it to obtain necessary and sufficient conditions for learnability. In doing so, we give a general learning algorithm that turns out to be surprisingly simple, though provably good. Section 3 deals with a slightly different learning model, one in which the learner is required to learn with one-sided error, i.e., his approximation to the set to be learned must be conservative in that it is a subset of the set to be learned. Section 4 deals with the time complexity of learning, identifying necessary and sufficient conditions for efficient learning. Section 5 generalizes the learning model to consider functions instead of sets. instead of sets. Notions of asymptotic learnability and asymptotic dimension are defined in this setting and necessary and sufficient conditions for learnability obtained. This requires us to prove a rather interesting combinatorial result called the generalized shattering lemma. Finally, Section 6 deals with a non-asymptotic model of learning, where the division is between finite and infinite, rather than on asymptotic behaviour. In

particular, we consider learning sets and functions on the reals, introducing the notion of finite-learnability. We review the elegant results of [Blumer et al., 1986] on conditions necessary and sufficient for learnability in this setting. We then identify conditions necessary and sufficient for the finite-learnability of functions on the reals.

2. Feasible Learnability of Sets

We begin by describing our variant of the learning framework proposed by [Valiant, 1984].

Let Σ be the binary alphabet (0,1), Σ^* the set of all strings on Σ, and for any positive integer n, let Σ^{n-} be the set of strings of length n or less in Σ^*. A *concept*[1] f is any subset of Σ^*. Associated with each concept f is the *membership function* $f^n:\Sigma \rightarrow \{0,1\}$, such that $f^n(x) = 1$ iff $x \in f$. Unless otherwise required, we will drop the superscript in f^n and use f to refer both to the function and to the set. An *example* for a concept is a pair (x,y), $x \in \Sigma^*$, $y \in \{0,1\}$ such that $y = f(x)$. A *family* of concepts F is any set of concepts on Σ^*. A *learning algorithm* (or more generally, a learning function) for the family F, is an algorithm that attempts to infer approximations to a concept in F from examples for it. The algorithm has at its disposal a subroutine EXAMPLE, which when called returns a randomly chosen example for the concept to be learned. The example is chosen randomly according to an arbitrary and unknown probability distribution P on Σ^*, in that the probability that a particular example $(x,f(x))$ will be produced at any call of EXAMPLE is $P(x)$.

Defn: Let f be a concept and n any positive integer. The projection f_n of f on Σ^{n-} is given by $f_n = f \cap \Sigma^{n-}$.

Defn: Let S be any set. A sequence on S is simply a sequence of elements of S. S^l denotes the set of all sequences of length l on S, while $\lambda(S)$ denotes the set of all sequences of finite length on S.

Defn: Let f be a concept on Σ^* and P a probability distribution on Σ^*. A *sample* of size l for f with respect to P is a sequence of the form $(x_1,f(x_1)), (x_2,f(x_2)),\ldots,(x_l,f(x_l))$ where $x_1, x_2,\ldots, x_l$ is a sequence of elements of Σ^*, randomly and independently chosen according to P.

Defn: Let f and g be any two sets. The symmetric difference of f and g, denoted by $f\Delta g$, is defined by $f\Delta g = (f-g)\cup(g-f)$.

With these supporting definitions in hand, we present our main definition. Intuitively, we will call a family F feasibly learnable if it can be learned from polynomially few examples, polynomial in an error parameter h and a length parameter n. The length parameter n controls the length of the strings the concept is to be approximated on, and the error parameter h controls the error allowed in the learnt approximation.

Defn: Formally, a family F is *feasibly learnable* if there exists an algorithm[2] A such that

(a) A takes as input two integers n and h, where n is the size parameter, and h is the error parameter.

(b) A makes polynomially few calls of EXAMPLE, polynomial in n and h. EXAMPLE returns examples for some $f \in F$, where the examples are chosen randomly and independently according

---

[1] we use the term concept instead of a set to conform with the artificial intelligence literature.

[2] Unless stated otherwise, by "algorithm" we mean a finitely representable procedure, not necessarily computable. That is, the procedure might use well-defined but non-computable functions as primitives.

to an arbitrary and unknown probability distribution P on $\Sigma^{n\infty}$.

(c) For all concepts $f \in F$ and all probability distributions P on $\Sigma^{n\infty}$, with probability $(1-1/h)$, A outputs a concept $g \in F$ such that

$$\sum_{x \in f \Delta g} P(x) \leq 1/h$$

Defn: Let **N** be the set of natural numbers. The *learning function* Ψ:**N**×**N**×$\lambda(\Sigma^* \times \{0,1\}) \rightarrow F$ associated with a learning algorithm A is defined as follows.

Learning Function Ψ
Input n, h:integers; C: sample;
begin
Let $C = (x_1, y_1), (x_2, y_2), \ldots$
Run A on inputs n,h;
In place of EXAMPLE, at the i^{th} call of EXAMPLE by A,
give A (x_i,y_i) as example.
Output A's output.
end

We now introduce a measure called the dimension for a family of concepts. Recall that we defined the projection f_n of f on Σ^n by $f_n = (f \cap \Sigma^n)$ Similarly, the projection F_n of the family F on Σ^n is given by $F_n = \{f_n | f \in F\}$. We call F_n the n^{th}-subfamily of F.

Defn: The *dimension* of a subfamily F_n, denoted by $dim(F_n)$ is defined by

$dim(F_n) = log_2(|F_n|)$.

(Notation: For a set X, $|X|$ denotes the cardinality, while for a string x, $|x|$ denotes the string length.)

Defn: Let d:**N** $\rightarrow$ **N** be a function of one variable, where **N** is the natural numbers. The *asymptotic dimension* (or more simply the dimension) of a family F is $d(n)$ if $dim(f_n) = \Theta(d(n))$. That is, there exists a constant c such that

$\forall\, n : dim(F_n) \leq d(n)$

and $dim(F_n) \geq cd(n)$ infinitely often.

We denote the asymptotic dimension of a family F by $dim(F)$. We say a family F is of polynomial dimension if the asymptotic dimension of F is a polynomial in n.

With these definitions in hand, we can give our first result. The result is a lemma concerning the notion of shattering. Let F be a family of subsets of set X. We say that F *shatters* a set $S \subseteq X$, if for every $S_1 \subseteq S$, there exists $f \in F$ such that $f \cap S = S_1$. To our knowledge, this notion was first introduced by [Vapnik & Chervonenkis, 1971].

We can now state our first result.

Lemma 1 (Shattering Lemma:) If F_n is of dimension d, then F_n shatters a set of size [3]$ceiling(d/(n+2))$. Also, every set shattered by F is of size at most d.

---

[3] $ceiling(r)$ is the least integer greater than r.

Proof: First, we prove the upper bound. Suppose a set S is shattered by by F_n. Since there are $2^{|S|}$ distinct subsets of F_n, it follows from the definition of shattering that $2^{|S|} \leq |F_n|$. Taking logarithms on both sides of the inequality, we get $|S| \leq log(|F_n|) = d$, which is as desired. To prove that the upper bound can attained, simply let F be all possible subsets of some d strings in Σ^{n-}.

We prove the lower bound part of the lemma through the following claim. A variant of the claim is given by Vapnik & Chervonenkis (1971) amongst others.

Claim: Let X be any finite set and let H be a set of subsets of X. If k is the size of the largest subset of X shattered by H, then

$|H| \leq (|X|+1)^k$.

Proof: By induction on $|X|$, the size of X.

Basis: Clearly true for $|X| = 1$.

Induction: Assume the claim holds for $|X| = m$ and prove true for $m+1$. Let $|X| = m+1$ and let H be any set of subsets of X. Also, let k be the size of the largest subset of X shattered by H. Pick any $x \in X$ and partition X into two sets $\{x\}$ and $Y = X-\{x\}$. Define H_1 to be the set of all sets in H that are reflected about x. That is, for each set h_1 in H_1, there exists a set $h \in H$ such that h differs from h_1 only in that h does not include x. Formally,

$H_1 = \{h_1 | h_1 \in H, \exists h \in H, h \neq h_1$ and $h_1 = h \cup \{x\}\}$.

Now define $H_2 = H-H_1$. Surely, the sets of H_2 can be distinguished on the elements of Y. That is, no two sets of H_2 can differ only on x, by virtue of our definition of H_1. Hence, we can consider H_2 as sets defined on Y. Surely, H_2 cannot shatter a set larger than the largest set shattered by H. Hence, H_2 shatters a set no bigger than k. Since $|Y| \leq m$, by the inductive hypothesis we have $|H_2| \leq (|Y|+1)^k$.

Now consider H_1. By definition, the sets of H_1 are all distinct on Y. That is, for any two distinct sets h_1, h_2 in H_1, $h_1 \cap Y \neq h_2 \cap Y$. Suppose H_1 shattered a set $S \subseteq Y$, $|S| \geq k$. Then, H would shatter $S \cup \{x\}$. But, $|S \cup \{x\}| \geq k+1$, which is impossible by assumption. Hence, H_1 shatters a set of at most $(k-1)$ elements in Y. By the inductive hypothesis, we have

$|H_1| \leq (|Y|+1)^{k-1}$.

Combining the two bounds, we have

$$\begin{aligned} |H| &= |H-H_1| + |H_1| = |H_2| + |H_1| \\ &\leq (|Y|+1)^k + (|Y|+1)^{k-1} \leq (m+1)^k + (m+1)^{k-1} \\ &\leq (m+1)^{k-1}(m+2) \leq (m+2)^k \leq (|X|+1)^k. \end{aligned}$$

Thus the claim is proved.•

Returning to the lemma, we see that if X is all strings of length n or less on the binary alphabet, $|X| = 2^{n+1}$. By our claim, if the largest set shattered by F_n is of size k,

$|F_n| \leq (2^{n+1}+1)^k$.

Hence, $k \geq log(|F_n|)/log(2^{n+1}+1) \geq dim(F_n)/(n+2)$.

Since k must be an integer, we take the ceiling of the right-hand side of the last inequality. This completes the proof of the lemma. •

We can now use this lemma to prove the main theorem of this section.

Theorem 1: A family F of concepts is feasibly learnable if and only if it is of polynomial dimension.

Proof: (If) Let F be of dimension $d(n)$. The following is a learning algorithm for F, satisfying the requirements of our definition of learnability.

Learning_Algorithm A_1

input: n, h
begin
call EXAMPLE $h(dim(F_n)ln(2) + ln(h))$ times.
let S be the set of examples seen.
pick any concept g in F consistent with S
output g.
end

We need to show that A_1 does indeed satisfy our requirements. Note that A_1 may not be computable, but, as noted earlier, this is not a difficulty. Let f be the concept to be learned. Since P is a distribution on Σ^n, EXAMPLE returns examples of f_n. We require that with high probability, A_1 should output a concept $g \in F$, such that the probability that f and g differ is less than $(1/h)$. Let $C_h(f)$ be all concepts in F_n that differ from f_n with probability greater than $1/h$. By definition, for any particular g such that $g_n \in C_h(f)$, the probability that any call of EXAMPLE will produce an example consistent with g is bounded by $(1-1/h)$. Hence, the probability that m calls of EXAMPLE will produce examples all consistent with g is bounded by $(1-1/h)^m$. And hence, the probability that m calls of EXAMPLE will produce examples all consistent with any $g_n \in C_h(f)$ is bounded by $|C_h(f)|(1-1/h)^m$. We wish to make m sufficiently large to bound this probability by $1/h$.

$|C_h(f)|(1-1/h)^m \leq 1/h$.

But surely, $|C_h(f)| \leq |F_n| \leq 2^{d(n)}$

Hence, we want

$2^{d(n)}(1-1/h)^m \leq 1/h$

Taking natural logarithms on both sides of the inequality, we get

$d(n)ln(2) + m \cdot ln(1-1/h) \leq ln(1/h)$

$-m \cdot ln(1-1/h) \geq d(n)ln(2) + ln(h)$

$-m\,(-1/h) \geq d(n)ln(2) + ln(h)$

Or

$m \geq h(d(n)ln(2)+ln(h))$.

Hence, if $h(d(n)ln(2)+ln(h))$ examples are drawn, the probability that all the examples seen are consistent with a concept that differs from the true concept by $1/h$ or more, is bounded by $1/h$. Since, A_1 draws as

many examples and outputs a concept consistent with the examples seen, with probability $1-1/h$, A_1 will output a concept that differs from the true concept with probability less than $1/h$. Hence, A_1 does satisfy our requirements. Clearly, if $d(n)$ is a polynomial in n, the number of examples called by A_1 is polynomial in n, h and hence F is feasibly learnable.

(only if)

Now suppose that F is of super-polynomial dimension $d(n)$ and yet F were feasibly learnable by an algorithm A from $(nh)^k$ examples, for some fixed k. Let Ψ be the learning function corresponding to A. Now pick n and $h \geq 5$ such that

$dim(F_n) \geq 2(n+1)(nh)^k$.

By the shattering lemma, there exists a set $S \subseteq \Sigma^n$ such that $|S| \geq dim(F_n)/(n+1)$, and S is shattered by F_n. Let $X^l \in S^l$ denote the sequence $x_1, x_2, \ldots, x_l$ and let $f \in F_n$. Define the operator δ as follows.

$$\delta(f, X^l, \Psi) = \sum_{x \in f \Delta g} P(x)$$

where $g = \Psi(n, h, (x_1 f(x_1)), (x_2 f(x_2)), \ldots (x_l f(x_l)))$,

In words, $\delta(f, X^l, \Psi)$ is the probability error in the concept output by A on seeing the sample $(x_1 f(x_1))$, $(x_2 f(x_2)), \ldots (x_l f(x_l))$ for f. Let $G_n \subseteq F_n$ be such that for each $S_1 \subseteq S$, there is exactly one $g \in G_n$ such that $g \cap S = S_1$. Such G_n must exist as F_n shatters S. Let P be the probability distribution that is uniform on S and zero elsewhere.

Claim: Let $l = (nh)^k$. Then for each $f \in G_n$, and $X^l \in S^l$, there exists unique $g \in G_n$ such that $\delta(f, X^l, \Psi) \leq 1/h$ if and only if $\delta(g, X^l, \Psi) \geq 1/h$.

Proof: Let $\{X^l\}$ denote the set of strings occuring in X^l, i.e., $\{X^l\} = \{x | x \text{ occurs in } X^l\}$. By the definition of G_n, for each f, X^l, there exists unique $g \in g_n$ such that $f \Delta g = S - \{X^l\}$. Hence,

$$\delta(f, X^l, \Psi) + \delta(g, X^l, \Psi) \geq \sum_{x \in S - \{X^l\}} P(x)$$

$$\geq 1/2.$$

The last step follows from the fact that $\{X^l\}$ has at most half as many elements as S, and P is uniform on S. Since $h \geq 5$, $1/h \leq 1/5$, at most one of the terms on the left can be smaller than (1/5), if the inequality is to hold. Hence the claim. •

Since Ψ is a learning function for F, for each $f \in F_n$

$\mathbf{Pr}\{\delta(f, X^l, \Psi) \leq 1/h)\} \geq (1-1/h)$

(Notation: $\mathbf{Pr}\{Y\}$ denotes the probability of event Y.)

Define the switch function θ:{true,false} $\rightarrow \mathbf{N}$ as follows. For any boolean-valued predicate Q,

$$\theta(Q) = \begin{cases} 1, \text{ if } Q \text{ is true} \\ 0 \text{ otherwise} \end{cases}$$

Now write

$$\mathbf{Pr}\{\delta(f, X^l, \Psi) \leq 1/h\} = \sum_{X^l \in S^l} \theta(\delta(f, X^l, \Psi) \geq 1/h) \mathbf{Pr}\{X^l\}$$

Substituting the above in the last inequality, we get,

$$\sum_{X^i \in S^i} \theta(\delta(f,X^i,\Psi) \geq 1/h)\mathbf{Pr}\{X^i\} \geq (1-1/h)$$

Summing over G_n,

$$\sum_{f \in G_n} \sum_{X^i \in S^i} \theta(\delta(f,X^i,\Psi) \geq 1/h)\mathbf{Pr}\{X^i\} \geq \sum_{f \in G_n} (1-1/h)$$

Flipping the order of the sums,

$$\sum_{X^i \in S^i} \sum_{f \in G_n} \theta(\delta(f,X^i,\Psi) \geq 1/h)\mathbf{Pr}\{X^i\} \geq \sum_{f \in G_n} (1-1/h)$$

By the last Claim,

$$\sum_{f \in G_n} \theta(\delta(f,X^i,\Psi) \geq 1/h)Pr(X^i) \leq \sum_{f \in G_n} (1/2)\mathbf{Pr}\{X^i\}$$

Hence, we have

$$\sum_{X^i \in S^i} \sum_{f \in G_n} (1/2)Pr(X^i) \geq \sum_{f \in G_n} (1-1/h)$$

Flipping the order of the sums again,

$$\sum_{f \in G_n} \sum_{X^i \in S^i} (1/2)Pr(X^i) \geq \sum_{f \in G_n} (1-1/h)$$

Which reduces to

$$\sum_{f \in G_n} (1/2) \geq \sum_{f \in G_n} (1-1/h)$$

which is impossible as $h \geq 5$.

The last contradiction implies that A cannot be a learning algorithm for F as supposed and hence the result.

This completes the proof. •

3. Learning Sets with One-Sided Error

We now consider a learning framework in which the learner is only allowed to see positive examples for the concept to be learned, and is required to be conservative in his approximation in that the concept output by the learner must be a subset of the concept to be learnt. Historically, this was the framework first studied by [Valiant, 1984].

Let F be the family of concepts to be learned. EXAMPLE produces positive examples for some concept $f \in F$. Specifically, EXAMPLE produces a string $x \in f$. Let P be a probability distribution on Σ^*. The probability that a string $x \in f$ is produced by any call of EXAMPLE is the conditional probability given by,

$$\frac{P(x)}{\sum_{x \in f} P(x)}$$

assuming the denominator is non-zero. If the denominator is zero, EXAMPLE never produces any examples. We can now define learnability as we did earlier.

Defn: A family of concepts F is *feasibly learnable with one-sided error* if there exists an algorithm A such that

(a) A takes as inputs integers n and h, where n is the size parameter and h the error parameter.

(b) A makes polynomially few calls of EXAMPLE, polynomial in n and h. EXAMPLE returns positive examples for some concept $f \in F$, chosen according to an arbitrary and unknown probability distribution P on Σ^{n-}.

(c) For all concepts $f \in F$ and all probability distributions P on Σ^{n-}, with probability $(1-1/h)$, A outputs $g \in F$ such that $g \subseteq f$ and

$$\sum_{x \in f \Delta g} P(x) \leq 1/h.$$

Defn: We say a family of concepts F is *well-ordered* if for all n, $F_n \cup \emptyset$ is closed under intersection.

With these definitions in hand, we state and prove the following theorem.

Theorem 2: A family F of concepts is feasibly learnable with one-sided error, if and only if it is of polynomial dimension and is well-ordered.

Proof: (If) This direction of the proof begins with the following claim.

Claim: Let $S \subseteq \Sigma^{n-}$ be any non-empty set such that there exists a concept $g \in F_n$ containing S. i.e. $g \in F_n$, and $S \subseteq g$. If F is well-ordered, there exists a *least* concept f in F_n containing g, i.e.,
$\forall\, g \in F_n$: $S \subseteq g$ implies $f \subseteq g$.

Proof: Let $S \subseteq \Sigma^{n-}$ be non-empty and let $\{f_1, f_2 \ldots\}$ be the set of concepts in F_n containing S. Now the intersection of all these concepts $f = \{f_1 \cap f_2 \cap \ldots\}$, is in F_n. To see this, notice that since $F_n \cup \emptyset$ is closed under intersection, $f \in F_n \cup \emptyset$. But, $f \neq \emptyset$ as $S \neq \emptyset$ and $S \subseteq f$. Hence, $f \in F_n$. •

This allows us to write the following learning algorithm for F.

Learning_Algorithm A_2

Input: n, h
begin
call EXAMPLE $h(d(n)\cdot ln(2) + ln(h))$ times.
let S be the set of examples seen.
output any g in F such that g_n is the least
concept in F_n containing S.
end

Let f be the concept to be learned. Since g_n is the least concept consistent with S, surely, $g_n \subseteq f_n$. Using arguments identical to those used in our proof of Theorem 1, we can show that with probability greater than $(1-1/h)$, g will not differ from the concept to be learned with probability greater than $1/h$. This completes the "if" direction of our proof.

(only if) Let F be feasibly learnable with one-sided error by an algorithm A. Let us show that F is well-ordered, i.e., for all n, $F_n \cup \emptyset$ is closed under intersection. Suppose for some n, $F_n \cup \emptyset$ were not closed under intersection, and that f, g were two concepts in $F_n \cup \emptyset$ such that $f \cap g$ is not in $F_n \cup \emptyset$. Now, surely $f \cap g \neq \emptyset$, and hence $f \cap g$ is not in F_n. Place the probability distribution that is uniform on $f \cap g$ and zero elsewhere on Σ^n, and run the learning algorithm A for $h = 2^{n+1}$. At each call of EXAMPLE, a randomly chosen element of $f \cap g$ will be returned. Since $f \cap g$ is not in F_n, A must fail to learn with one-sided error. To see this, suppose that A outputs some concept $e \in F$. Now, since A claims to learn with one sided error, $e_n \subseteq f$, if f were the concept to be learned. Similarly, $e_n \subseteq g$, since g could well be the concept to be learned. Hence, $e_n \subseteq f \cap g$. But since $h=1/2^{n+1}$, e_n must be $f \cap g$, which contradicts the assumption that $f \cap g$ is not in F_n. By arguments similar to those of our proof of Theorem 1, we can show that F must be of polynomial dimension. An alternate proof is presented in [Natarajan, 1986]. Hence the claim. •

This completes the proof. •

We now exhibit a curious property of the well-ordered families. Specifically, we show that each concept (except the empty set) in a well-ordered family has a short and unique "signature".

For a well ordered family F, define the operator $M_n: 2^{\Sigma^n} \rightarrow F_n$ as follows.

$$M_n(S) = \begin{cases} \text{least } f \in F_n \text{ such that } S \subseteq f, \text{ if such } f \text{ exists} \\ \text{undefined otherwise} \end{cases}$$

In words, $M_n(S)$ is simply the least set in F_n consistent with S.

Proposition 1: M_n is idempotent, i.e.,

$M_n(M_n(S)) = M_n(S)$

Proof: By the definition of M_n, $M_n(S)$ is the least concept $f \in F_n$ such that $S \subseteq f$. Surely, $M_n(f) = f$ and hence the proposition. •

Proposition 2: For $A \subseteq B \subseteq \Sigma^n$, if $M_n(A)$ and $M_n(B)$ are both defined, then

$M_n(A) \subseteq M_n(B)$.

Proof: By the definition of M_n, $B \subseteq M_n(B)$. Since $A \subseteq B$, $A \subseteq M_n(B)$. Hence, $M_n(A) \subseteq M_n(B)$, by Proposition 1. •

Proposition 3: For $A, B \subseteq \Sigma^*$, if $M_n(A)$ and $M_n(B)$ are defined,

$M_n(A \cup B) = M_n(M_n(A) \cup M_n(B))$

Proof: Since $A \subseteq M_n(A)$, $B \subseteq M_n(B)$, $A \cup B \subseteq M_n(A) \cup M_n(B)$. Whence it follows from Proposition 2 that, $M_n(A \cup B) \subseteq M_n(M_n(A) \cup M_n(B))$. And then, since $A \subseteq A \cup B$, we have by Proposition 2

$M_n(A) \subseteq M_n(A \cup B)$

and similarly

$M_n(B) \subseteq M_n(A \cup B)$

Hence,

$M_n(A) \cup M_n(B) \subseteq M_n(A \cup B)$

Applying Proposition 2 again, we get

$M_n(M_n(A) \cup M_n(B)) \subseteq M_n(M_n(A \cup B))$

Applying Proposition 1 to the right-hand side,

$M_n(M_n(A) \cup M_n(B)) \subseteq M_n(A \cup B)$.

Hence, the proposition. •

With these supporting propositions in hand, we can show that every concept in F has a small "signature".

Proposition 4: If F is well-ordered, then for every $f \in F_n$, $f \neq \emptyset$ there exists $S_f \subseteq \Sigma^{n-}$, $|S_f| \leq dim(F_n)$, such that $f = M_n(S_f)$.

Proof: Let $f \in F_n$ and let S_f be a set of minimum size such that $f = M_n(S_f)$. Consider any two distinct subsets S_1, S_2 of S_f. We claim that $M_n(S_1) \neq M_n(S_2)$. To prove this, we will assume the contrary and arrive at a contradiction. Suppose $M_n(S_1) = M_n(S_2)$ for $S_1 \neq S_2$. Without loss of generality, assume $|S_1| \leq |S_2|$. Now,

$S_f = (S_f - S_2) \cup S_2$

Applying M_n to both sides,

$M_n(S_f) = M_n((S_f - S_2) \cup S_2)$

Applying Proposition 2 to the right-hand side, we get

$M_n(S_f) = M_n(M_n(S_f - S_2) \cup M_n(S_2))$

Since $M_n(S_2) = M_n(S_1)$,

$M_n(S_f) = M_n(M_n(S_f - S_2) \cup M_n(S_1))$

Applying Proposition 2 again,

$M_n(S_f) = f = M_n((S_f - S_2) \cup S_1)$

But $|(S_f - S_2) \cup S_1| < |S_f|$,

which contradicts our assumption that S_f was a set of minimum size such that $f = M_n(S_f)$. Hence, each distinct subset of S_f corresponds to a distinct $f \in F_n$. (Notice that we have really shown that S_f is

shattered by F_n.) Which in turn implies that

$|F_n| \geq 2^{|S_f|}$

or

$dim(F_n) \geq |S_f|$

Hence the proposition. •

Conversely, we can show that Proposition 4 is tight in the following sense.

Proposition 5: If F is well-ordered, there exists $f \in F_n$ such that $f = M_n(S)$ implies $|S| \geq dim(F_n)/(n+1)$.

Proof: A simple counting argument. There are at most 2^{n+1} distinct examples. If every $f \in F_n$ were definable as the least concept containing some set of d examples, then

$2^{(n+1)d} \geq |F_n|$ or

$(n+1)d \geq dim(F_n)$ implying $d \geq dim(F_n)/(n+1)$.

Hence, the proposition. •

4. Time-Complexity Issues in Learning Sets

Thus far, we concerned ourselves with the information complexity of learning, i.e.,, the number of examples required to learn. Another issue to be considered is the time-complexity of learning, i.e., the time required to process the examples. In order to permit interesting measures of time-complexity, we must specify the manner in which the learning algorithm identifies its approximation to the unknown concept. In particular, we will require the learning algorithm to output a name of its approximation in some predetermined naming scheme. To this end, we define the notion of an index for a family of concepts.

In order for each concept in a family F to have a name of finite length, F would have to be at most countably infinite. Assuming that the family F is countably infinite, we define an *index* of F to be a function $I:F \rightarrow 2^{\Sigma^*}$ such that

$\forall f,g \in F, f \neq g$ implies $I(f) \cap I(g) = \emptyset$.

For each $f \in F$, $I(f)$ is the set of indices for f.

We are primarily interested in families that can be learnt efficiently, i.e., in time polynomial in the input parameters n, h and in the length of the shortest index for the concept to be learned. Analogous to our definition of learnability, we can now define polynomial-time learnability as follows. Essentially, a family is polynomial-time learnable, if it is feasibly learnable by a polynomial-time algorithm.

Defn; A family of concepts F is *polynomial-time learnable* in an index I if there exists a deterministic learning algorithm A such that

(a) A takes as input integers n and h.

(b) A runs in time polynomial in the error parameter h, the length parameter n and in the length of the shortest index in I for the concept to be learned f. A makes polynomially few calls of EXAMPLE, polynomial[4] in n, h. EXAMPLE returns examples for f chosen randomly according to an arbitrary and unknown probability distribution P on Σ^{n}.

(c) For all concepts f in F and all probability distributions P on Σ^{n}, with probability $(1-1/h)$ the algorithm outputs an index $i_g \in I(g)$ of a concept g in F such that

$$\sum_{x \in f \Delta g} P(x) \leq 1/h$$

We are interested in identifying the class of pairs (F, I), where F is a family of concepts and I is an index for it, such that F is polynomial-time learnable in I. To this end, we define the following.

Defn: For a family F and index I, an *ordering* is a program that

(a) takes as input a set of examples $S = \{(x_1,y_1), (x_2,y_2),...(x_i,y_i..\}$ such that

$x_1, x_2, x_3... \in \Sigma^*$, and $y_1, y_2.. \in \{0,1\}$.

(b) produces as output an index in I of a concept $f \in F$ that is consistent with S, if such exists. i.e., outputs $i_f \in I(f)$ for some $f \in F$ such that

$\forall (x,y) \in S,\ y = f(x)$.

---

[4]Alternatively, we could permit A to make as many calls of EXAMPLE as possible within its time bound. This will not change our discussion substantially. In the interest of clarity we will not pursue this alternative.

Furthermore, if the ordering runs in time polynomial in the length of its input and the length of the shortest such index, we say it is a polynomial-time ordering and F is *polynomial-time orderable* in I.

With these definitions in hand, we can state the following theorem.

Theorem 3: A family of concepts is polynomial-time learnable in an index I (1) if it is of polynomial dimension and is polynomial-time orderable in I. (2) only if F is of polynomial dimension and is random polynomial time orderable in I.[5]

Proof: (If) Let Q be a polynomial-time ordering for F in I. The following is a polynomial time learning algorithm for F in I.

Learning_Algorithm A_3

Input: n, h
begin
call EXAMPLE $h(dim(F_n) + log(h))$ times;
let S be the set of examples seen;
output $Q(S)$;
end

Given Theorem 1, we know that A_3 learns F, and only need bound its running time polynomial. Now, Q runs in time polynomial in the size of its input and the length of the shortest index of any concept consistent with S. Since the concept to be learned must be consistent with S, surely Q runs in time polynomial in n, h and in the length of the shortest index of the the concept to be learned. Hence, A_3 runs in time polynomial in n, h and in the length of the shortest index for the concept to be learned. Therefore, F is polynomial-time learnable in I.

(Only if) Assume that F is polynomial time learnable in an index I by an algorithm A. Since A calls for polynomially few examples, F must be of polynomial dimension by Theorem 1. It remains to show that there exists a randomized polynomial-time ordering for F. The following is such an ordering.

Ordering O
Input: S:set of examples, n:integer;

begin
place the uniform distribution on S;
let $h = |S|+1$:
run A on inputs n, h, and
on each call of EXAMPLE by A
return a randomly chosen element of S.
output the index output by A.
end

Let f be a concept consistent with S, whose index length is the shortest over all such concepts. Now, with probability $(1-1/h)$ A must output the index of a concept g that agrees with f with probability greater

[5]A randomized algorithm is one that tosses coins during its computation and produces the correct answer with high probability.

than $(1-1/h)$. Since the distribution is uniform and $h > |S|$, g must agree with f on every example in S. Hence with high probability, g is consistent with S. Furthermore, since A is a polynomial-time learning algorithm for F, our ordering O is a randomized polynomial-time ordering for F in I. To see this, notice that A runs in time polynomial in n and h, and l, the length of the shortest index of f. By our choice of h, it follows that A runs in time polynomial in n, $|S|$ and l. Hence, O runs in time polynomial in n, h and l, and is a randomized polynomial-time ordering for F in I.

This completes the proof. •

We can state analogous results on the time-complexity of learning with one-sided error. Specifically, an ordering for a well-ordered family would be an ordering as defined earlier with the exception that it would produce the least concept consistent with the input. Also, we can modify our definition of polynomial time learnability to allow only one-sided error. We can then state and prove the following.

Theorem 4: A family F is polynomial-time learnable with one-sided error; (1) if it is of polynomial dimension, well-ordered and possesses a polynomial time ordering; (2) only if it is of polynomial dimension, well-ordered and possesses a random polynomial time ordering.

Proof: A straightforward extension of earlier proofs. •

5. Learning Functions

In the foregoing, we were concerned with learning approximations to concepts or sets. In the more general setting, one may consider learning functions from Σ^* to Σ^*. To do so, we must first modify our definitions suitably and generalize our formulation of the problem.

Defn: We define a family of functions to be any set of functions from Σ^* to Σ^*. For any $f \in F_n$, the projection $f_n: \Sigma^n \rightarrow \Sigma^n$ of f on Σ^n is given by

$$f_n(x) = \begin{cases} f(x), \text{ if } |f(x)| = n \\ n\text{–length prefix of } x, \text{ otherwise} \end{cases}$$

Defn: The n^{th}-subfamily F_n of F is the projection of F on Σ^n, i.e,

$F_n = \{f_n | f \in F\}$.

The above two definitions are the analogues of the corresponding definitions for sets. The notion of the projection f_n of a function f attempts to capture the behaviour of f on strings of length n. If for some $x \in \Sigma^{n-}$, $f(x)$ is not of length at most n, it is truncated to n characters.

An example for a function f is a pair (x,y), $x,y \in \Sigma^*$ such that $y = f(x)$. A learning algorithm (or more precisely a learning function) for a family of functions is an algorithm that attempts to infer approximations to functions in F from examples for it. The learning algorithm has at its disposal a subroutine EXAMPLE, which at each call produces a randomly chosen example for the function to be learned. The examples are chosen according to an arbitrary and unknown probability distribution P in that the probability that a particular example $(x,f(x))$ will be produced at any call is $P(x)$.

As in the case of sets, we define learnability as follows.

Defn: A family of functions F is *feasibly learnable* if there exists an algorithm A such that

(a) A takes as input integers n and h, where n is the size parameter and h the error parameter.

(b) A makes polynomially few calls of EXAMPLE, polynomial in n and h. EXAMPLE returns examples for some function $f_n \in F_n$, chosen according to an arbitrary and unknown probability distribution P on Σ^{n-}.

(c) For all functions $f_n \in F_n$ and all probability distributions P on Σ^{n-}, with probability $(1-1/h)$, A outputs a a function $g \in F$ such that

$$\sum_{f_n(x) \neq g_n(x)} P(x) \leq 1/h$$

Our definition of dimension in this setting is exactly the same as the one given earlier for concepts. We can now generalize the notion of shattering as follows.

Defn: Let F be a family of functions from a set X to a set Y. We say F *shatters* a set $S \subseteq X$ if there exist two functions $f, g \in F$ such that

(a) for any $s \in S$, $f(s) \neq g(s)$.

(b) for all $S_1 \subseteq S$, there exist $e \in F$ such that e agrees with f on S_1 and with g on $S-S_1$. i.e.,

$\forall s \in S_1: e(s) = f(s)$
$\forall s \in S-S_1: e(s) = g(s)$.

We can now generalize our shattering lemma for functions as follows.

Lemma 2 (Generalized Shattering Lemma): If F_n is of dimension d, F_n shatters a set of size $ceiling(d/(3n+3)))$. Also, every set shattered by F_n is of size at most d.

Proof: The upper bound part of the lemma can be proved exactly as the corresponding part of Lemma 1. To see that this upper bound can be attained, we simply need to consider a family F_n of $\{0,1\}$-valued functions.

The lower bound part of the lemma is proved through the following claim.

Claim: Let X and Y be two finite sets and let H be a set of functions from X to Y. If k is the size of the largest subset of X shattered by H, then
$|H| \leq (|X|)^k(|Y|)^{2k}$.

Proof: By induction on $|X|$.

Basis: Clearly true for $|X| = 1$, for all $|Y|$.

Induction: Assume true for $|X| = l$, $|Y| = m$ and prove true for $|X| = l+1$, $|Y| = m$. Let $X = \{x_1, x_2 \ldots, x_l\}$ and $Y = \{y_1, y_2 \ldots, y_l\}$. Define the subsets H_i of H as follows.
$H_i = \{f \mid f \in H, f(x_1) = y_i\}$.

Also, define the sets of functions H_{ij} and H_0 as follows.
for $i \neq j$: $H_{ij} = \{f \mid f \in H_i, \exists g \in H_j$ such that $f = g$ on $X-\{x_1\}\}$.
$H_0 = H - \bigcup_{i \neq j} H_{ij}$.

Now,

$$|H| = |H_0| + |\bigcup_{i \neq j} H_{ij}| \leq |H_0| + \sum_{i \neq j} |H_{ij}|.$$

We seek bounds on the quantities on the right-hand side of the last inequality. By definition, the functions in H_0 are all distinct on the m elements of $X-\{x_1\}$. Furthermore, the largest set shattered in H_0 must be of cardinality no greater than k. Hence, we have by the inductive hypothesis,
$|H_0| \leq l^k m^{2k}$.
And then, every H_{ij} shatters a set of cardinality at most $k-1$, as otherwise H would shatter a set of cardinality greater than k. Also, since the functions in H_{ij} are all distinct on $X - \{x_1\}$, we have by the inductive hypothesis,
For $i \neq j$, $|H_{ij}| \leq l^{k-1} m^{2(k-1)}$.

Combining the last three inequalities, we have

$$|H| \leq l^k m^{2k} + \sum_{i \neq j} l^{k-1} m^{2(k-1)} \leq l^k m^{2k} + m^2 l^{k-1} m^{2(k-1)} \leq l^k m^{2k} + l^{k-1} m^{2k}$$

$$\leq m^{k-1} l^{2k} (m+1) \leq (m+1)^k l^{2k}.$$

Which completes the proof of the claim. •

Returning to the lemma, we have $X = Y = \Sigma^{n+1}$, and hence $l = m = 2^{n+1}$. If k is the cardinality of the largest set in Σ^{n+1} shattered by F_n, we have by our claim,

$$|F_n| \leq (2^{n+1})^k (2^{n+1})^{2k}$$
$$\leq 2^{k(3n+3)}.$$

Taking logarithms,

$$\log(|F_n|) = dim(F_n) = d \leq k(3n+3)$$

Hence, $k \geq d/(3n+3)$, which is as desired. •.

Using this lemma, we can prove the following theorem.

Theorem 5: A family of functions is feasibly learnable if and only if it is of polynomial dimension.

Proof: Similar to the proof of Theorem 1, except that we need use the generalized notion of shattering and the corresponding generalized shattering lemma. •

Analogous to our development of time-complexity considerations for concept learning, we define the following.

For a family of functions F of countable cardinality, we define an index I to be a naming scheme for the functions in F, in a sense identical to that for a family of concepts.

We say a family of functions F is *polynomial-time learnable* in an index I, if there exists a deterministic learning algorithm A such that

(a) A takes as input integers n and h.

(b) A runs in time polynomial in the error parameter h, the length parameter n and in the length of the shortest index in I for the function to be learned f. A makes polynomially few calls of EXAMPLE, polynomial in n, h. EXAMPLE returns examples for f_n chosen randomly according to an arbitrary and unknown probability distribution P on Σ^n.

(c) For all concepts f in F and all probability distributions P on Σ^n, with probability $(1-1/h)$ the algorithm outputs an index $i_g \in I(g)$ of a function g in F such that

$$\sum_{f_n(x) \neq g_n(x)} P(x) \leq 1/h$$

We are interested in identifying the class of pairs (F, I), where F is a family of concepts and I is an index for it, such that F is polynomial-time learnable in I. To this end, we define the following.

Defn: For a family F and index I, an *ordering* is a program that

(a) takes as input a set of examples $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i) \ldots\}$. Let n be the length of the longest string among the x_i and y_i.

(b) produces as output an index in I of a concept $f \in F$ that is consistent with S, if such exists. i.e., outputs $i_f \in I(f)$ for some $f \in F$ such that

$\forall (x,y) \in S,\ y = f_n(x)$.

Furthermore, if the ordering runs in time polynomial in the length of its input and the length of the shortest such index, we say it is a polynomial-time ordering and F is *polynomial-time orderable* in I.

With these definitions in hand, we can state the following theorem.

Theorem 6: A family of functions is polynomial-time learnable: (1) if it is of polynomial dimension and polynomial-time orderable; (2) only if it is of polynomial dimension and is orderable in random polynomial time.

Proof: Similar to that of Theorem 3. •

6. Finite Learnability

Thus far we explored the asymptotic learnability of families of sets and functions, that is to say, we considered the asymptotic variation of the number of examples needed for learning with increasing values of the size parameter. We will now investigate a different notion of learnability, one that asks whether the number of examples needed for learning is finite, i.e, varies as a finite-valued function of the error parameter, without regard to the size parameter. We call this notion of learnability "finite learnability" as opposed to the notion of asymptotic learnability.

For the case of families of sets, [Blumer et al., 1986] present conditions necessary and sufficient for finite-learnability. Their elegant results rely on the powerful results in classical probability theory of [Vapnik and Chervonenkis, 1971]. In the following we review their results briefly and then go on to present learnability results for families of functions, relying in part on the same results of [Vapnik and Chervonenkis, 1971].

Defn: Let F be a family of sets on $\mathbf{R}^k$, where $\mathbf{R}$ is the set of reals and k is a fixed natural number. We say F is *finitely learnable* if there exists an algorithm A such that

(a) A takes as input integer h, the error parameter.

(b) A makes finitely many calls of EXAMPLE, although the exact number of calls may depend on h. EXAMPLE returns examples for some function f in F, where the examples are chosen randomly according to an arbitrary and unknown probability distribution P on $\mathbf{R}$.

(c) For all probability distributions P and all functions f in F, with probability $(1-1/h)$, A outputs $g \in$ F

$$\int_{f \neq g} dP \leq 1/h$$

The following theorem is from [Blumer et al., 1986].

Theorem 7: [Blumer et al., 1986] A family of sets F on $\mathbf{R}^k$ is finitely learnable if and only if F shatters only finite subsets of $\mathbf{R}^k$. ([Blumer et al., 1986] refer to the size of the largest set shattered by F as the *Vapnik-Chervonenkis dimension* of the family F).

Let us now formalize the notion of finite learnability of families of functions on the reals.

Defn: Let F be a family of functions from $\mathbf{R}^k$ to $\mathbf{R}^k$, where $\mathbf{R}$ is the set of reals and k is a fixed natural number. We say F is *finitely learnable* if there exists an algorithm A such that

(a) A takes as input integer h, the error parameter.

(b) A makes finitely many calls of EXAMPLE, although the exact number of calls may depend on h. EXAMPLE returns examples for some function f in F, where the examples are chosen randomly according to an arbitrary and unknown probability distribution P on $\mathbf{R}^k$.

(c) For all probability distributions P and all functions f in F, with probability $(1-1/h)$, A outputs $g \in F$ such that

$$\int_{f \neq g} dP \leq 1/h$$

We need the following supporting definitions. Let f be a function from $\mathbf{R}^k$ to $\mathbf{R}^k$. We define the *graph* of f, denoted by *graph*(f), to be the set of all examples for f. That is,

$graph(f) = \{(x,y) | y = f(x)\}$.

Clearly, $graph(f) \subseteq \mathbf{R}^k \times \mathbf{R}^k$. Analogously, for a family of functions F, we define $graph(F)$ to be the set of graphs for the functions in F. That is,

$graph(F) = \{graph(f) | f \in F\}$.

We now state the main theorem of this section. The theorem is not tight in the sense that the necessary and sufficient conditions do not match. (In [Natarajan, 1988], a tight version of the theorem was reported, on the basis of an incorrect proof.) Indeed, we will identify a finitely learnable family of functions that sits in the gap between these conditions.

Theorem 8: A family of functions F from $\mathbf{R}^k$ to $\mathbf{R}^k$ is finitely learnable

(a) If there exists a bound on the size of the sets in $\mathbf{R}^k \times \mathbf{R}^k$ shattered by $graph(F)$. (simple shattering as defined in Section 2.)

(b) Only if there exists a bound on the size of the sets in $\mathbf{R}^k$ shattered by F. (Generalized shattering as defined in Section 5.)

Proof: (If) This direction of the proof follows from the convergence results of [Vapnik and Chervonenkis, 1971] exactly as shown in [Blumer et al., 1986]. Essentially, the "if" condition implies that the family $graph(f)$ is finitely learnable. Whence it follows that the family F is finitely learnable.

(Only if) This direction of the proof is identical to the asymptotic case of Theorem 4, which in turn followed the arguments of Theorem 1. •

While Theorem 8 is not tight, it appears that tightening it is a rather difficult task. Indeed we conjecture that the "if" condition should match the "only if" condition as stated below.

Conjecture: A family of functions F from $\mathbf{R}^k$ to $\mathbf{R}^k$ is finitely learnable if and only if there exists a bound on the size of the sets in $\mathbf{R}^k$ shattered by F.

To give the reader a flavour of the difficulties involved in tightening Theorem 8, we give an example of a family F of functions that lies in the gap between the necessary and sufficient conditions of Theorem 8, i.e

(a) F shatters sets of size at most one.

(b) $graph(F)$ shatters arbitrarily large sets.

(c) F is finitely learnable.

Example: Let $\mathbf{N}$ be the natural numbers in binary representation. For any $\alpha \in \mathbf{N}$, define the function $f_\alpha: \mathbf{N} \rightarrow \mathbf{N}$ as follows.

$$f_\alpha(x) = \begin{cases} \alpha, \text{ if the } x^{th} \text{ bit of } \alpha \text{ is } 1 \\ 0 \text{ otherwise} \end{cases}.$$

Define the family F as follows.

$F = \{f_\alpha | \alpha \in \mathbf{N}\}$.

Claim: F shatters sets of size at most one.

Proof: Suppose F shatters a set of size greater than one. Then F must shatter a set of size 2. Let $S = \{a,b\}$ be such a set. By definition, there exist three functions f, g, e in F such that $f(a) \neq g(a), f(b) \neq g(b)$ and $e(a) = f(a)$, $e(b) = g(b)$. Since, $f(a) \neq g(a)$, one of them must be zero and the other non-zero. Without loss of generality, assume that $f(a)$ is non-zero. Now, by the definition of the functions in F, $f(a) = e(a) \neq 0$ implies that $f = e$. This contradicts the assumption that $e(b) = g(b) \neq f(b)$, and hence the claim. •

Claim: $graph(F)$ shatters arbitrarily large sets.

Proof: Let S_1 be any arbitrarily large but finite subset of **N**. Consider $S = S_1 \times \{0\}$. It is easy to see that $graph(F)$ shatters S, as for any subset S_2 of S, there exists a set $f \in F$ such that $f \cap S = S_2$. To see this, notice that for any subset S_2 of S, we can pick an integer $\alpha \in$ **N**, such that $f_\alpha \cap S = S_2$. Since S was picked to be arbitrarily large, the claim is proved. •

Claim: F is finitely learnable.

Proof: The following is a learning algorithm for F.

Learning Algorithm A_4
input h;

begin
call for $hlog(h)$ examples.
If any of the examples seen is of the
form (x,y), $y \neq 0$
then output f_y
else output f_0.
end

It is easy to show that the probabilities work out for algorithm A above. Suppose the function to be learned were f_α, for some $\alpha \neq 0$. Then, if

$$\int_{f_\alpha \neq f_0} dP \geq 1/h,$$

with probability $(1-1/h)$, in $hlogh$ examples there must be an example of the form (x,α). In which case, the algorithm will output f_α, implying that with probability $(1-1/h)$, the algorithm learns the unknown function exactly. Hence the claim. •

The interesting thing about the functions in F is that each function differs from the base function f_0 on finitely many points, and on these points, the value of the function is the name of the function. Hence, if the learning algorithm sees a non-zero value in an example, it can uniquely identify the function be learned. •

Thus far, we considered functions on real spaces, requiring that on a randomly chosen point, with high probability the learner's approximation agree exactly with the function to be learned. This requires

infinite precision arithmetic and hence is largely of technical interest. But then, if all the computations are carried out only to some finite precision, Theorem 5 would apply directly. Alternatively, we could require that the learned function approximate the target function with respect to some predetermined norm. In the following, we consider the case of the square norm, for a single probability distribution P.

First, we limit the discussion to families of "normalized" functions. Let $E(a,b)$ denote the euclidean distance between any two points a and b. Let $F:\mathbf{R}^k \rightarrow \mathbf{R}^k$ be a family of functions such that for every $f \in F$ and $x \in \mathbf{R}^k$, $E(f(x),0^k) \leq 1$, where 0^k is the origin in $\mathbf{R}^k$. Then, we fix the probability distribution P.

Defn: We say that F is finitely learnable with respect to the square norm and a distribution P on $\mathbf{R}^k$, if there exists an algorithm A such that:

(a) A takes as input an integer h, the error parameter.

(b) A makes finitely many calls of EXAMPLE, though the exact number may depend on h. EXAMPLE returns examples for some function f in F, where the examples are chosen according to the distribution P.

(c) For all functions $f \in F$, with probability h, A outputs a function $g \in F$ such that

$$\int_{x \in \mathbf{R}^k} E(f(x),g(x))dP \leq 1/h.$$

Before we can state our result in this setting, we need the following definition, adapted from [Benedeck and Itai, 1988].

Defn: For small positive δ: $K \subseteq F$ is a δ-cover with respect to the square norm and distribution P if, for any $f \in F$ there exists $g \in K$ such that,

$$\int_{x \in \mathbf{R}^k} E(f(x),g(x))dP \leq \delta$$

Theorem 9: A family of functions is finitely learnable with respect to the square norm and a distribution P, if and only if for all positive δ, there exists a finite δ-cover for F.

Proof: The details of the proof are identical to that of the main theorem of [Benedeck and Itai, 1988]. A learning algorithm A for F can be described as follows: on input h, A constructs an $1/h$-cover of F of minimum size. A then calls for sufficiently many examples to permit it to pick one of the functions in the knot with sufficiently high confidence. •

7. Acknowledgements

I thank R.Kannan, D.Sleator, P.Tadepalli and T.Mitchell for many useful discussions.

8. References

[1] Angluin, D., and Smith, C.H., (1983). Inductive Inference: Theory and Methods. *Computing Surveys, 15(3.* (pp. 237-269).

[2] Angluin, D., (1986). Learning Regular Sets from Queries and Counter-Examples, Tech. Report, YALEU/DCS/TR-464.

[3] Benedeck, G.M., and Itai, N., (1988). Learning by Fixed Distributions. *Proceedings of the Workshop on Computational Learning Theory.* (pp. 80-90).

[4] Berman, P., and Roos, R., (1987). Learning One-Counter Languages in Polynomial Time. In *Proceedings of the Symposium on Foundations of Computer Science.* (pp. 61-67).

[5] Blumer, A., Ehrenfeucht, A, Haussler, D., & Warmuth, M., (1986). Classifying Learnable Geometric Concepts with the Vapnik-Chervonenkis Dimension., In *Proceedings of the ACM Symposium on Theory of Computing* (pp. 273-282).

[6] Kearns, M., Li, M., Pitt, L., and Valiant, L.G., (1987). On the Learnability of Boolean Formulae. In *Proceedings of the ACM Symposium on Theory of Computing.* (pp. 285-295).

[7] Laird, P., (1987) Learning from Data Good and Bad, Ph.D Thesis, Dept. of Computer Science, Yale University.

[8] Michalski, R., Mitchell, T., and Carbonell, J., *Machine Learning: An Artificial Intelligence Approach*, Tioga Press, Palo Alto, CA.

[9] Natarajan, B.K., (1986). On Learning Boolean Functions. Carnegie-Mellon Robotics Institute TR-86-17 and in *Proceedings of the ACM Symposium on Theory of Computing, 1987.* (pp. 296-304).

[10] Natarajan, B.K., (1988). Two New Frameworks for Learning, *Proceedings of the Fifth International Symposium on Machine Learning.* (pp. 402-415).

[11] Rivest, R., and Schapire, R.E., (1987) Diversity Based Inference of Finite Automata. In *Proceedings of the Symposium on Foundations of Computer Science.* (pp. 78-87).

[13] Valiant, L.G., (1984). A Theory of the Learnable. In *Proceedings of the ACM Symposium on Theory of Computing.* (pp.436-445).

[14] Vapnik, V.N., and Chervonenkis, A.Ya., (1971). On the Uniform Convergence of Relative Frequencies, Theory of Probability and its Applications, vol16, No2, (pp.264-280).